

CPU and memory usage.

Memstat tool

Outlook

- We need standard tool to measure the Memory and CPU consumption
 - AliSysInfo – to combine together CPU and Mem usage information
- Memstat library in the AliRoot
 - Extended functionality in comparison with old ROOT memory checker

AliSysInfo

- Store base information about Process (code already committed to the CVS)
- Written to the separate log file
- Information per stamp
 - Current memory usage
 - Current timer
 - Stamp name + 3 indexes
- Example usage in AliReconstruction:
 - `AliSysInfo::AddStamp(Form("LRec%s_%d",fgkDetectorName[iDet],eventNr), iDet,1,eventNr);`

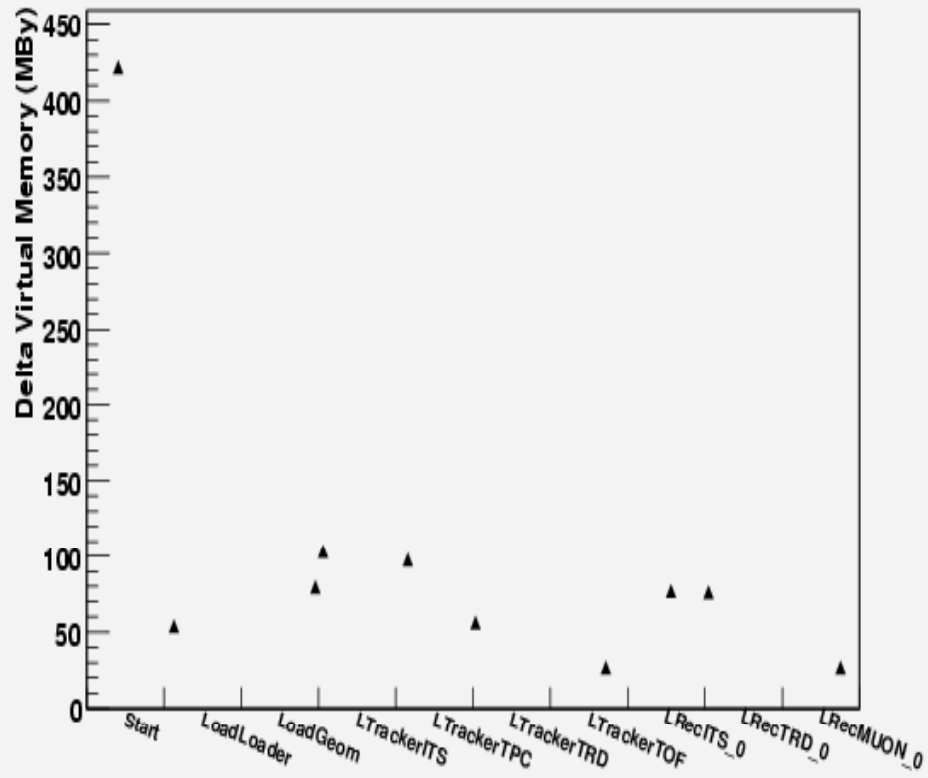
AliSysInfo report

- The report generated
 - Top n (example 10) violators in memory consumption
 - Top n violators for CPU consumption
 - Reports per detector

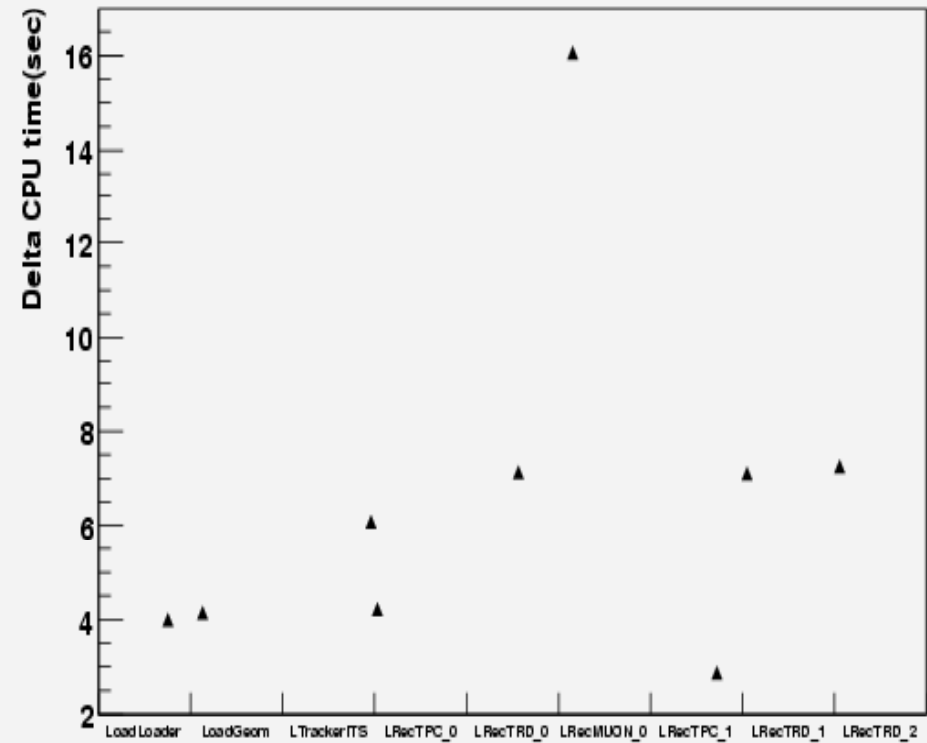
```
gROOT->LoadMacro("$ALICE_ROOT/macros/PlotSys.C+");  
MakePlots("syswatch.log","syswatch.root",10);  
TFile f("syswatch.root");  
TBrowser b;
```

Report - LowMultiplicity

deltaVM:sname {(1)&&(deltaVM>23.351999)}

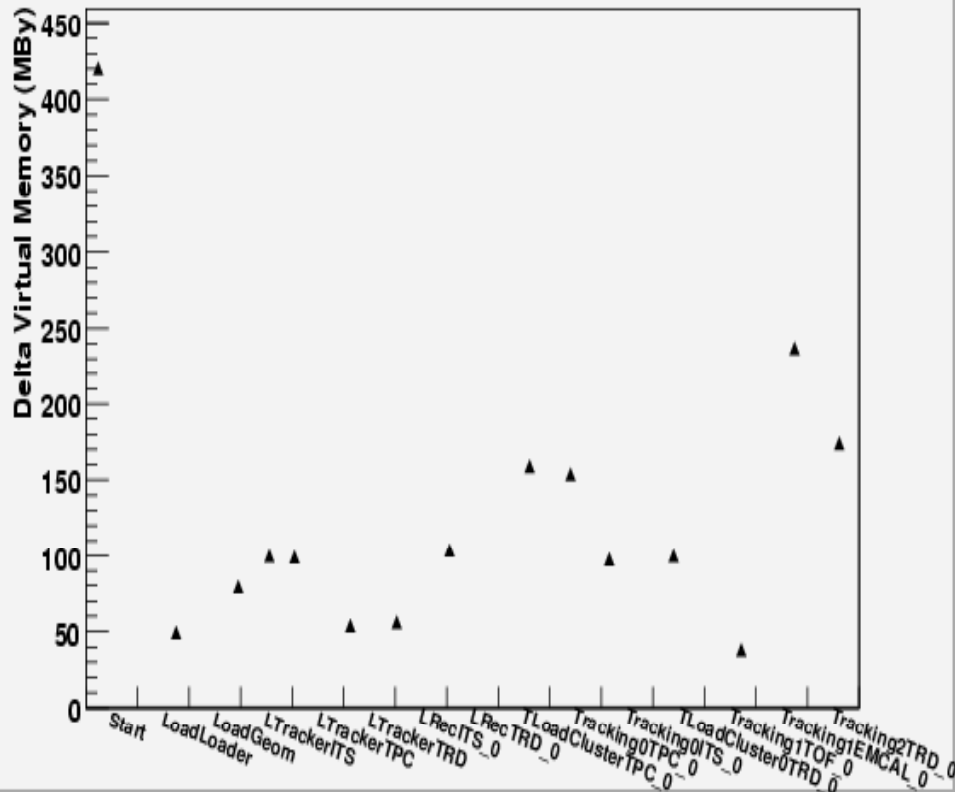


deltaT:sname {(id2<3)&&(deltaT>2.000000)}

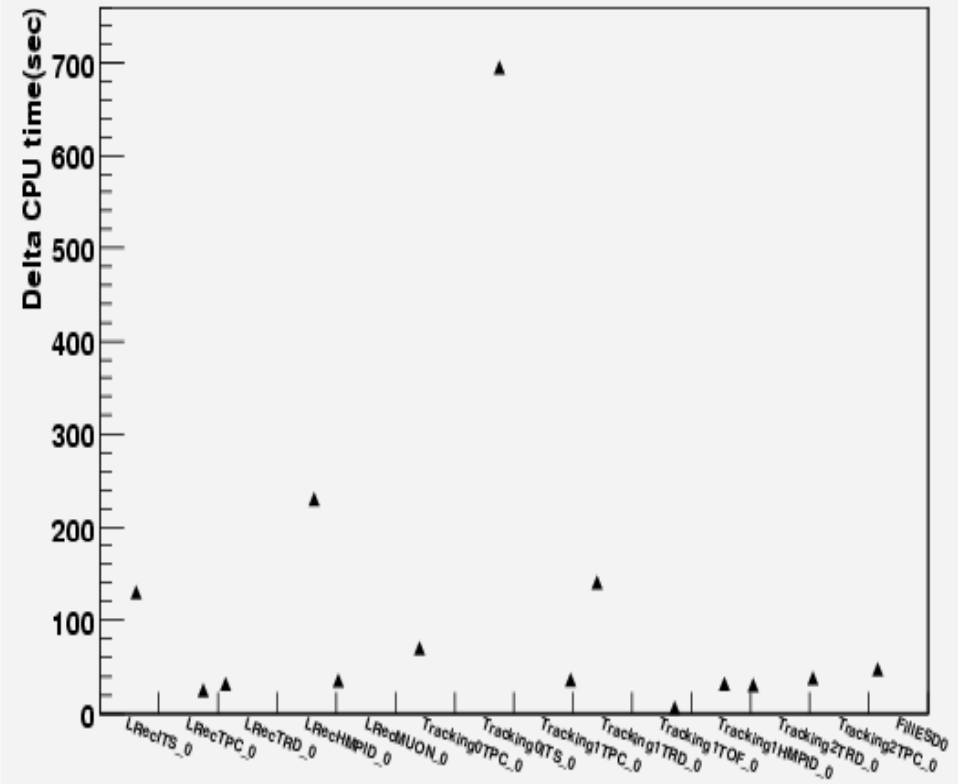


Report – High Multiplicity

deltaVM:sname {(1)&&(deltaVM>37.967999)}



deltaT:sname {(id2<3)&&(deltaT>16.000000)}

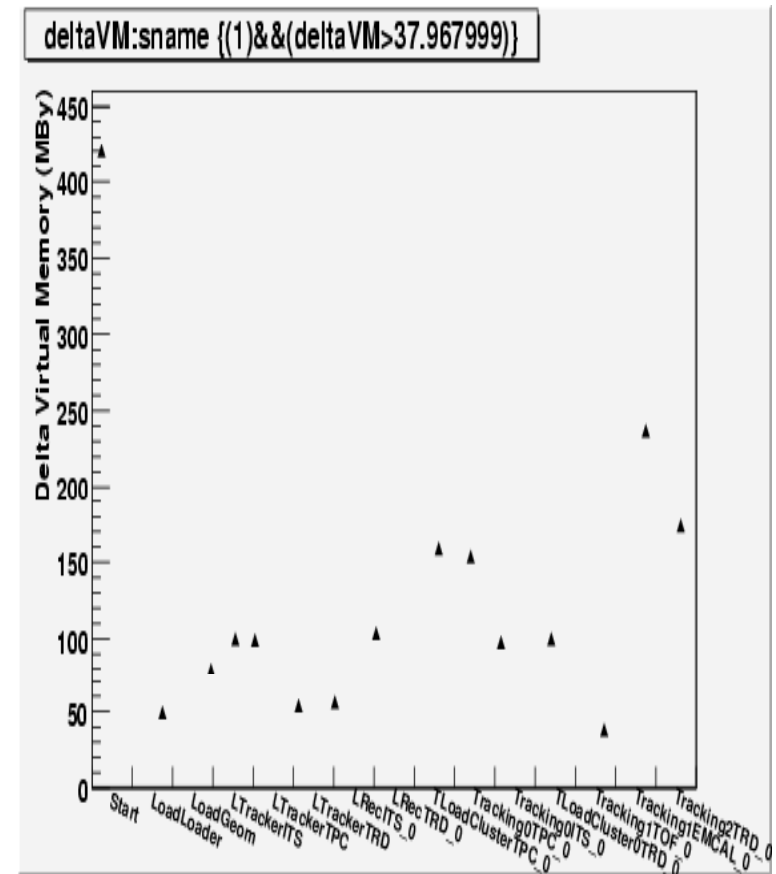


Summary CPU

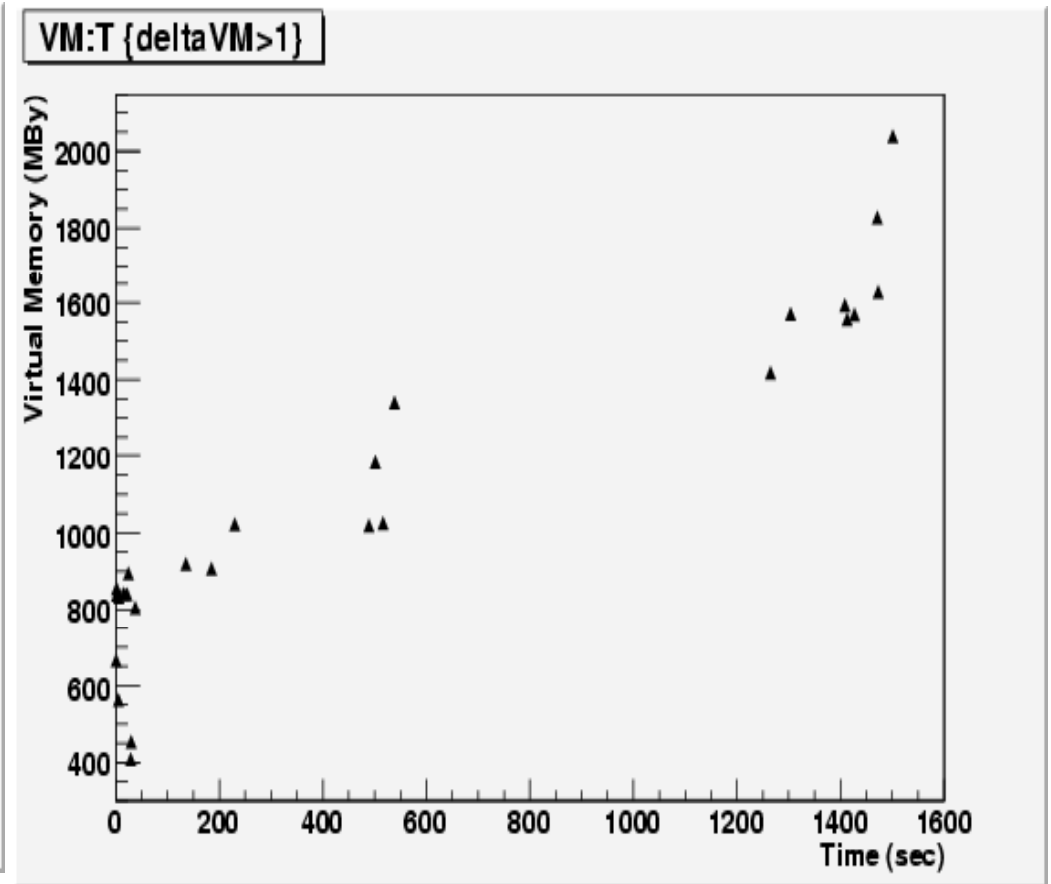
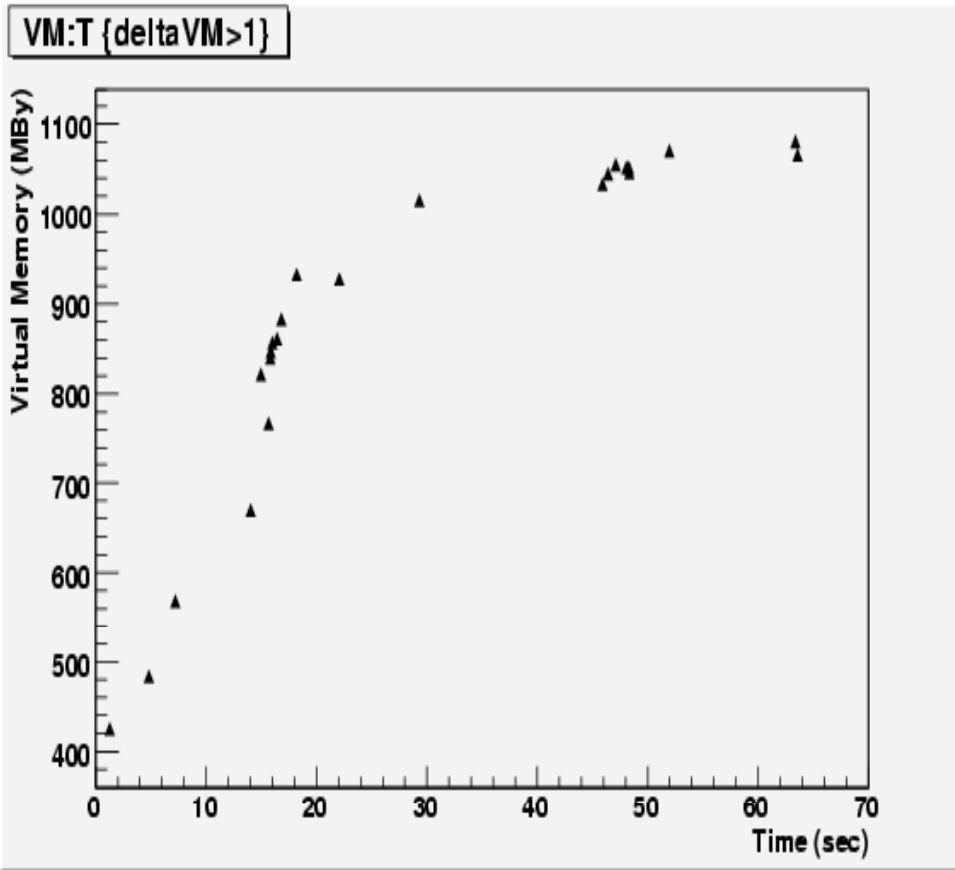
- The CPU time is currently determined by the ITS tracker
 - Usage of the TgeoManager should be optimized
- The CPU per TPC tracker ~ 50 s
- Other outliers
 - ITS local reconstruction
 - TRD tracking
 - HMPID local reconstruction

Summary – Memory consumption

- 2GBy reached ~450 Mby at start of AliRoot
- TPC part didn't change (~300MBy)
 - 160 Mby -load clusters
 - 160MBy creates TPC and ESDClusters



Memory usage



Memory usage.

- Spikes at the end of the reconstruction
 - Virtual Memory is not returning back to the system

```
* Row * Mem3-CurrentStamp.fA * Mem3 * CurrentStamp.fAllocS * StampName.String() *
*****
*****
* 0 * 497.769954427734376 * 498.420013427734375 * 0.65005900000000053 * Start *
* 1 * 613.961552703124994 * 671.719970703125 * 57.7584179999999989 * LoadLoader *
* 2 * 614.042043113281238 * 671.80401611328125 * 57.7619729999999976 * LoadCDB *
* 3 * 624.727837628906286 * 721.23199462890625 * 96.50415700000000064 * LoadGeom *
* 4 * 624.751473445312513 * 721.2559814453125 * 96.50450800000000013 * Vertexer *
* 5 * 624.693314511718768 * 876.61199951171875 * 251.9186850000000011 * LoadTrackers *
* 6 * 633.564376320312476 * 972.0040283203125 * 338.4396520000000024 * LRecTPC_0 *
* 7 * 713.505303054687488 * 1057.0799560546875 * 343.5746530000000012 * LRecTRD_0 *
* 8 * 712.209567054687568 * 1057.0799560546875 * 344.8703889999999989 * RLoadClusterTPC_0 *
* 9 * 632.623190656250017 * 1231.27197265625 * 598.6487819999999983 * TLoadClusterTPC_0 *
* 10 * 702.697554148437462 * 1469.9759521484375 * 767.27839800000000038 * Tracking0TPC_0 *
* 11 * 700.658902148437505 * 1469.9759521484375 * 769.3170499999999995 * Tracking1TPC_0 *
* 12 * 700.599620148437452 * 1469.9759521484375 * 769.37633200000000048 * RLoadCluster0TRD_0 *
* 13 * 664.489779640624988 * 1653.008056640625 * 988.5182770000000012 * TLoadCluster0TRD_0 *
* 14 * 666.350684828125054 * 1671.300048828125 * 1004.9493639999999995 * Tracking1TRD_0 *
* 15 * 675.286122914062389 * 1870.7359619140625 * 1195.4498390000000011 * Tracking2TRD_0 *
* 16 * 877.472130914062518 * 1870.7359619140625 * 993.2638309999999982 * TUnloadClusterTRD_0 *
* 17 * 877.734729351562464 * 1870.3599853515625 * 992.62525600000000036 * RUnloadClusterTRD_0 *
* 18 * 850.014209351562499 * 1870.3599853515625 * 1020.345776 * Tracking2TPC_0 *
* 19 * 1103.1222733515624 * 1870.3599853515625 * 767.2377119999999988 * TUnloadClusterTPC_0 *
* 20 * 1105.08823535156262 * 1870.3599853515625 * 765.2717499999999997 * RUnloadClusterTPC_0 *
* 21 * 1103.00354335156248 * 1870.3599853515625 * 767.3564420000000015 * FilIESD0 *
* 22 * 1640.64902628906248 * 1849.8199462890625 * 209.1709199999999995 * End *
* 23 * 1640.64902628906248 * 1849.8199462890625 * 209.1709199999999995 * End *
*****
*****
```